IPMI Firmware User Guide

for the

AM4100

Module

Manual ID: 36725 Rev. Index 1.0
Firmware: SK-IPMI-AM4100, EKS Idx 0100
October 15, 2007

kontron

1. Copyright

Copyright © 2007 Kontron AG

Kontron Modular Computers makes no representations or warranties with respect to the contents or use of this manual, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Kontron Modular Computers makes no representations or warranties with respect to this embedded Linux package, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Permission is granted to make and distribute verbatim copies of this manual provided that the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this documentation under the conditions for verbatim copying, provided also that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this documentation into another language, under the above conditions for modified versions.

The PICMG® and CompactPCI® names and the PICMG®, CompactPCI®, ATCA®, and AdvancedTCA® logos are registered trademarks and AdvancedMC is a trademark of the PCI Industrial Computer Manufacturers Group.

Intel is a registered trademark of Intel Corporation.

I^2C is a trademark of Phillips Semiconductors.

Linux is a registered trademark of Linus Torvalds.

All other trademarks, registered trademarks, and trade names are the property of their respective owners.

2. Revision History

Manual/Product Title:	IPMI Firmware User Guide for the AM4100 Module	
Manual ID Number:	36725	
Revision Index	**Brief Description of Changes**	**Date of Issue**
1.0	Initial Issue	Oct. 15, 2007

Imprint

Kontron Modular Computers GmbH may be contacted via the following:

MAILING ADDRESS

Kontron Modular Computers GmbH

Sudetenstraße 7

D - 87600 Kaufbeuren Germany

TELEPHONE AND E-MAIL

+49 (0) 800-SALESKONTRON

sales@kontron.com

For further information about other Kontron products, please visit our Internet web site: www.kontron.com

Disclaimer

Copyright © 2007 Kontron AG. All rights reserved. All data is for information purposes only and not guaranteed for legal purposes. Information has been carefully checked and is believed to be accurate; however, no responsibility is assumed for inaccuracies. Kontron and the Kontron logo and all other trademarks or registered trademarks are the property of their respective owners and are recognized. Specifications are subject to change without notice.

3. Contents

1. Copyright 2
2. Revision History 3
3. Contents 4
4. Introduction 6
4.1 Acronym Definitions 6
4.2 Related Documentation 7
5. Introduction 8
5.1 IPMI in AdvancedMC / AdvancedTCA Environment 8
5.2 Module Management Controller Hardware 8
6. MMC Firmware 9
6.1 Key Features 9
6.2 Supported IPMI Commands 10
6.2.1 Standard Commands 10
6.2.2 OEM Commands and Extensions 17
6.2.2.1 Get Device ID Command with OEM Extensions 17
6.2.2.2 Set Firmware Parameters 18
6.2.2.3 Set Control State (Firmware Hub, Boot Order) 19
6.2.2.4 Get Control State (Firmware Hub, Boot Order) 20
6.2.2.5 Set or Get FRU Sub Device Number 21
6.2.2.6 OEM Module Quiescence Feedback 22
7. Sensors Implemented on the AM4100 24
7.1.1 Sensor List 25
7.1.2 OEM Event / Reading Types 28
8. Firmware Code 30
8.1 Structure and Functionality 30
8.2 Firmware Upgrade 30
8.3 Firmware / Module Identification 31
8.4 FRU Information 31
8.4.1 Structure and Functionality 31
8.4.2 Board Specific FRU Data Organization 32
8.4.3 Download 32

 © 2007 Kontron AG

8.5 E-Keying 32
8.6 PCI Express Clock Source 33
9. NetBootLoader Flash Selection 34
9.1.1 Boot Flash Selection by OEM IPMI Command 34
9.1.2 Automatic Boot Flash Selection during the Boot Process 34
10. Hot Swap 35
11. Module Not Running on an ATCA Carrier 36
12. OS Support / Tools 37
12.1.1 Linux 37
13. IPMI Communication LEDs 38

 © 2007 Kontron AG

4. Introduction

4.1 Acronym Definitions

AMC	Advanced Mezzanine Card
BSP	Board Support Package
DMI	Desktop Management Interface
FRU	Field Replaceable Unit
FWH	Firmware Hub
I²C	Inter-Integrated Circuit
IPMB	Intelligent Platform Management Bus
IPMB-0	AdvancedTCA shelf-level IPMB
IPMB-L	Local, on-carrier IPMB that links the carrier IPMC with the MMCs of installed modules
IPMC	Intelligent Platform Management Controller located on AMC carrier
IPMI	Intelligent Platform Management Interface
IOL	IPMI over LAN. An MMC is accessed via LAN, not IPMB
KCS	Keyboard Controller Style
MMC	Module Management Controller – an IPMI controller located on the AMC module
MP	Management Power
PICMG	PCI Industrial Computer Manufacturer Group
PWR	Payload Power
SDR	Sensor Data Record
SDRR	Sensor Data Record Repository
SEL	System Event Log
SMBIOS	System Management BIOS
SMS	System Management Software (designed to run under the OS)
SOL	Serial over LAN. A serial interface is redirected by LAN using the RMCP+ protocol.

© 2007 Kontron AG

4.2 Related Documentation

IPMI specifications: (http://www.intel.com/design/servers/ipmi/spec.htm)

- IPMI-Intelligent Platform Management Interface Specification. Second Generation v2.0, February 12, 2004 (part)
- [2] IPMI- Platform Management FRU Information Storage Definition v1.0, Document Revision 1.1, September 1999

PICMG specifications: http://www.picmg.org

- PICMG® AMC.0 R1.0 - Advanced Mezzanine Card Base Specification
- PICMG® AMC.1 R1.0 - PCI Express and Advanced Switching on AdvancedMC
- PICMG® AMC.2 R1.0 – AMC Gigabit Ethernet/10 Gigabit XAUI Ethernet

Open tools documentation

- [1] Ipmitool documentation: http://ipmitool.sourceforge.net
- OpenIPMI documentation: http://www.openipmi.sourceforge.net

Kontron manuals and specifications: http://www.kontron.com/

- AM4100 User's Guide
- [3] AM4100 Linux Board Support Package

As a hot-swappable field replaceable unit (FRU), the AM4100 follows the stringent carrier grade RASM feature set, namely - Reliability, Availability, Serviceability, Maintainability.

Built in accordance to the AMC.0 specification, the AM4100 is also AMC.1 and AMC.2 compliant and is easily managed via IPMI v1.5/v2.0.

As with every Advanced Mezzanine Card (AMC) the AM4100 is equipped with a Module Management Controller (MMC).

© 2007 Kontron AG

5. Introduction

5.1 IPMI in AdvancedMC / AdvancedTCA Environment

The Module Management Controller is a crucial component of any AMC module. Besides acting as a regular IPMI management controller (sensor monitoring, event logging, etc.), it also provides an interface to all necessary data related to module power requirements and implemented interfaces (E-Keying). Further, it plays an active role in the module hot swap state management. The carrier IPMI Controller (IPMC) communicates with the MMC using the local IPMB (IPMB-L) bus. In an ATCA/AMC environment, it is the IPMC that actually turns on/off module (payload) power. However, before the IPMC enables the module payload power, various criteria must be satisfied by both the carrier and the module, including power requirements and capabilities, matching interfaces, current module hot swap state, and any other special conditions as specified by the Shelf Manager policy.

5.2 Module Management Controller Hardware

On the AM4100 module, the MMC is implemented using the Renesas H8S/2166 controller with 512 kB of internal flash and 40 kB of RAM. An additional 1 MB serial EEPROM chip provides redundant firmware image storage. A separate 32 kB serial EEPROM chip is used for firmware private data and 4 kB FRU Inventory storage.

The MMC circuit implements two local Keyboard Style Interfaces (KCS) with interrupt support for communication with system side management software and the IPMB-L bus for interconnection with the IPMC.

The MMC implements a wide range of sensors that permit the monitoring of:

- main power voltages: +12V (PWR), +5V, +3.3V, +3.3V (MP), +1.5V, +1.8V, +0.9V (DDR) , +1.1V
- temperatures: CPU die, board inlet and outlet near CPU, board inlet and outlet near AMC connector
- Power Good, IPMB link, board reset, post code, boot error, processor hot detection (>105 °C), IPMB-L state, Health error, IPMI watchdog etc.

 © 2007 Kontron AG

6. MMC Firmware

6.1 Key Features

- compliant with the related IPMI and PICMG® specifications (refer to 4.2, *Related Documentation*)
- firmware designed and specially made for AdvancedMC environments
- supports two KCS interfaces with interrupt support
- supports the local IPMB (IPMB-L) interface
- out of band management and monitoring using IPMB-L interface permits access to sensors regardless of module CPU state
- sensor thresholds fully configurable
- sensor names prefixed with AMC module Bay ID (A1…4, B1…4)
- Usable in µTCA slots 1…12. Sensor names for slots 9…12 are prefixed with R9…R12
- complete IPMI watchdog functionality
- complete FRU functionality (refer to 8.4)
- firmware can be updated in the field (refer to 8.2)
- two firmware banks implemented, firmware bank management is done by the open tool ipmitool (function fwum) [1] (refer to 8.1)
- downloading new firmware image does not break currently running firmware activities (refer to 8.2).
- manual and automatic firmware image roll-back (in case of upgrade failure). Refer to 8.2.
- firmware customizable per boot loader or per OEM IPMI commands
- interoperable with other AMC, ATCA, or IPMI solutions
- OEM board supervision and control extensions such as boot device flash selection and firmware boot order configuration (refer to 6.2.2)
- Boot loader supervisor and automated switch to recover the boot image from a second boot device flash (refer to 9)
- Graceful shutdown support (refer to 10)
- “Health” LED shows heartbeat and pulses if a KCS interface is active. The “out-of-service” (OOS) LED pulses when IPMB-L is active (refer to 13).
- If the carrier supplies an external clock 3 the module will use it. Otherwise the module will use the internal PCIe clock (refer to 8.6).

© 2007 Kontron AG

6.2 Supported IPMI Commands

6.2.1 Standard Commands

Part of the command list in IPMI specification 2.0

M = mandatory, O = optional

	IPMI 2.0 Spec. section	NetFn	CMD	Kontron support On MMC
IPM Device "Global" Commands				**M**
Get Device ID	20.1	App	01h	M / **Yes [3]**
Cold Reset	20.2	App	02h	O / **Yes**
Warm Reset	20.3	App	03h	O / No
Get Self Test Results	20.4	App	04h	O / **Yes**
Manufacturing Test On	20.5	App	05h	O / No
Set ACPI Power State	20.6	App	06h	O / No
Get ACPI Power State	20.7	App	07h	O / No
Get Device GUID	20.8	App	08h	O / No
Broadcast "Get Device ID"	20.9	App	01h	M / **Yes**
BMC Watchdog Timer Commands				**O**
Reset Watchdog Timer	27.5	App	22h	O / **Yes**
Set Watchdog Timer	27.6	App	24h	O / **Yes**
Get Watchdog Timer	27.7	App	25h	O / **Yes**

© 2007 Kontron AG

BMC Device and Messaging Commands				O
Set BMC Global Enables	22.1	App	2Eh	O / **Yes**
Get BMC Global Enables	22.2	App	2Fh	O / **Yes**
Clear Message Flags	22.3	App	30h	O / **Yes**
Get Message Flags	22.4	App	31h	O / **Yes**
Enable Message Channel Receive	22.5	App	32h	O / **Yes**
Get Message	22.6	App	33h	O / **Yes**
Send Message	22.7	App	34h	O / **Yes**
Read Event Message Buffer	22.8	App	35h	O / **Yes**
Get BT Interface Capabilities	22.9	App	36h	O / **Yes**
Get System GUID	22.14	App	37h	O / No
Get Channel Authentication Capabilities	22.13	App	38h	O / **Yes**
Get Session Challenge	22.15	App	39h	O / **Yes**
Activate Session	22.17	App	3Ah	O / **Yes**
Set Session Privilege Level	22.18	App	3Bh	O / **Yes**
Close Session	22.19	App	3Ch	O / **Yes**
Get Session Info	22.20	App	3Dh	O / **Yes**
Get AuthCode	22.21	App	3Fh	O / No
Set Channel Access	22.22	App	40h	O / **Yes**
Get Channel Access	22.23	App	41h	O / **Yes**
Get Channel Info	22.24	App	42h	O / **Yes**
Set User Access	22.26	App	43h	O / **Yes**
Get User Access	22.27	App	44h	O / **Yes**
Set User Name	22.28	App	45h	O / **Yes**
Get User Name	22.29	App	46h	O / **Yes**
Set User Password	22.30	App	47h	O / **Yes**
Activate Payload	24.1	App	48h	O / **Yes**
Deactivate Payload	24.2	App	49h	O / **Yes**
Get Payload Activation Status	24.4	App	4Ah	O / **Yes**
Get Payload Instance Info	24.5	App	4Bh	O / **Yes**
Set User Payload Access	24.6	App	4Ch	O / **Yes**
Get User Payload Access	24.7	App	4Dh	O / **Yes**
Get Channel Payload Support	24.8	App	4Eh	O / **Yes**
Get Channel Payload Version	24.9	App	4Fh	O / **Yes**
Get Channel OEM Payload Info	24.10	App	50h	O / No
Master Write-Read	22.11	App	52h	O / **Yes**
Get Channel Cipher Suits	22.15	App	54h	O / No
Suspend/Resume Payload Encryption	24.3	App	55h	O / **Yes**
Set Channel Security Keys	22.25	App	56h	O / No
Get System Interface Capabilities	22.9	App	57h	O / No

© 2007 Kontron AG

Chassis Device Commands				**O / Yes**
Get Chassis Capabilities	28.1	Chassis	00h	O / **Yes**
Get Chassis Status	28.2	Chassis	01h	O / **Yes**
Chassis Control	28.3	Chassis	02h	O / No
Chassis Reset	28.4	Chassis	03h	O / No
Chassis Identify	28.5	Chassis	04h	O / No
Set Chassis Capabilities	28.7	Chassis	05h	O / No
Set Power Restore Policy	28.8	Chassis	06h	O / No
Get System Restart Cause	28.11	Chassis	07h	O / No
Set System Boot Options	28.12	Chassis	08h	O / No
Get System Boot Options	28.13	Chassis	09h	O / No
Get POH Counter	28.14	Chassis	0Fh	O / **Yes** **[1]**
Event Commands				**M**
Set Event Receiver	29.1	S/E	01h	M / **Yes**
Get Event Receiver	29.2	S/E	02h	M / **Yes**
Platform Event (a.k.a. “Event Message”)	29.3	S/E	03h	M / **Yes**
PEF and Alerting Commands				**O**
Get PEF Capabilities	30.1	S/E	10h	O / No
Arm PEF Postpone Timer	30.2	S/E	11h	O / No
Set PEF Configuration Parameters	30.3	S/E	12h	O / No
Get PEF Configuration Parameters	30.4	S/E	13h	O / No
Set Last Processed Event ID	30.5	S/E	14h	O / No
Get Last Processed Event ID	30.6	S/E	15h	O / No
Alert Immediate	30.7	S/E	16h	O / No
PET Acknowledge	30.8	S/E	17h	O / No

© 2007 Kontron AG

Sensor Device Commands				**M**
Get Device SDR Info	35.2	S/E	20h	M / **Yes**
Get Device SDR	35.3	S/E	21h	M / **Yes**
Reserve Device SDR Repository	35.4	S/E	22h	M / **Yes**
Get Sensor Reading Factors	35.5	S/E	23h	O / No
Set Sensor Hysteresis	35.6	S/E	24h	O / **Yes**
Get Sensor Hysteresis	35.7	S/E	25h	O / **Yes**
Set Sensor Threshold	35.8	S/E	26h	O / **Yes**
Get Sensor Threshold	35.9	S/E	27h	O / **Yes**
Set Sensor Event Enable	35.10	S/E	28h	O / **Yes**
Get Sensor Event Enable	35.11	S/E	29h	O / **Yes**
Re-arm Sensor Events	35.12	S/E	2Ah	O / No
Get Sensor Event Status	35.13	S/E	2Bh	O / No
Get Sensor Reading	35.14	S/E	2Dh	M / **Yes**
Set Sensor Type	35.15	S/E	2Eh	O / No
Get Sensor Type	35.16	S/E	2Fh	O / No
FRU Device Commands				**M**
Get FRU Inventory Area Info	34.1	Storage	10h	M / **Yes**
Read FRU Data	34.2	Storage	11h	M / **Yes**
Write FRU Data	34.3	Storage	12h	M / **Yes**
SDR Device Commands				**O**
Get SDR Repository Info	33.9	Storage	20h	O / No
Get SDR Repository Allocation Info	33.10	Storage	21h	O / No
Reserve SDR Repository	33.11	Storage	22h	O / No
Get SDR	33.12	Storage	23h	O / No
Add SDR	33.13	Storage	24h	O / No
Partial Add SDR	33.14	Storage	25h	O / No
Delete SDR	33.15	Storage	26h	O / No
Clear SDR Repository	33.16	Storage	27h	O / No
Get SDR Repository Time	33.17	Storage	28h	O / No
Set SDR Repository Time	33.18	Storage	29h	O / No
Enter SDR Repository Update Mode	33.19	Storage	2Ah	O / No
Exit SDR Repository Update Mode	33.20	Storage	2Bh	O / No
Run Initialization Agent	33.21	Storage	2Ch	O / No

© 2007 Kontron AG

SEL Device Commands				**O**
Get SEL Info	40.2	Storage	40h	O / No
Get SEL Allocation Info	40.3	Storage	41h	O / No
Reserve SEL	40.4	Storage	42h	O / No
Get SEL Entry	40.5	Storage	43h	O / No
Add SEL Entry	40.6	Storage	44h	O / No
Partial Add SEL Entry	40.7	Storage	45h	O / No
Delete SEL Entry	40.8	Storage	46h	O / No
Clear SEL	40.9	Storage	47h	O / No
Get SEL Time	40.10	Storage	48h	O / No
Set SEL Time	40.11	Storage	49h	O / No
Get Auxiliary Log Status	40.12	Storage	5Ah	O / No
Set Auxiliary Log Status	40.13	Storage	5Bh	O / No
LAN Device Commands				**O**
Set LAN Configuration Parameters	23.1	Transport	01h	O / **Yes**
Get LAN Configuration Parameters	23.2	Transport	02h	O / No
Suspend BMC ARPs	23.3	Transport	03h	O / **Yes**
Get IP/UDP/RMCP Statistics	23.4	Transport	04h	O / **Yes**
Serial/Modem Device Commands				**O**
Set Serial/Modem Configuration	25.1	Transport	10h	O / No
Get Serial/Modem Configuration	25.2	Transport	11h	O / No
Set Serial/Modem Mux	25.3	Transport	12h	O / No
Get TAP Response Codes	25.4	Transport	13h	O / No
Set PPP UDP Proxy Transmit Data	25.5	Transport	14h	O / No
Get PPP UDP Proxy Transmit Data	25.6	Transport	15h	O / No
Send PPP UDP Proxy Packet	25.7	Transport	16h	O / No
Get PPP UDP Proxy Receive Data	25.8	Transport	17h	O / No
Serial/Modem Connection Active	25.9	Transport	18h	O / No
Callback	25.10	Transport	19h	O / No
Set User Callback Options	25.11	Transport	1Ah	O / No
Get User Callback Options	25.12	Transport	1Bh	O / No
SOL Activating	26.1	Transport	20h	O / **Yes**
Get SOL Configuration Parameters	26.2	Transport	21h	O / **Yes**
Set SOL Configuration Parameters	26.3	Transport	22h	O / **Yes**

© 2007 Kontron AG

Bridge Management Commands (ICMB)				**O**
Get Bridge State	[ICMB]	Bridge	00h	O / No
Set Bridge State	[ICMB]	Bridge	01h	O / No
Get ICMB Address	[ICMB]	Bridge	02h	O / No
Set ICMB Address	[ICMB]	Bridge	03h	O / No
Set Bridge Proxy Address	[ICMB]	Bridge	04h	O / No
Get Bridge Statistics	[ICMB]	Bridge	05h	O / No
Get ICMB Capabilities	[ICMB]	Bridge	06h	O / No
Clear Bridge Statistics	[ICMB]	Bridge	08h	O / No
Get Bridge Proxy Address	[ICMB]	Bridge	09h	O / No
Get ICMB Connector Info	[ICMB]	Bridge	0Ah	O / No
Get ICMB Connection ID	[ICMB]	Bridge	0Bh	O / No
Send ICMB Connection ID	[ICMB]	Bridge	0Ch	O / No
Discovery Commands (ICMB)				**O**
Prepare For Discovery	[ICMB]	Bridge	10h	O / No
Get Addresses	[ICMB]	Bridge	11h	O / No
Set Discovered	[ICMB]	Bridge	12h	O / No
Get Chassis Device ID	[ICMB]	Bridge	13h	O / No
Set Chassis Device ID	[ICMB]	Bridge	14h	O / No
Bridging Commands (ICMB)				**O**
Bridge Request	[ICMB]	Bridge	20h	O / No
Bridge Message	[ICMB]	Bridge	21h	O / No
Event Commands (ICMB)				**O**
Get Event Count	[ICMB]	Bridge	30h	O / No
Set Event Destination	[ICMB]	Bridge	31h	O / No
Set Event Reception State	[ICMB]	Bridge	32h	O / No
Send ICMB Event Message	[ICMB]	Bridge	33h	O / No
Get Event Destination	[ICMB]	Bridge	34h	O / No
Get Event Reception State	[ICMB]	Bridge	35h	O / No
OEM Commands for Bridge NetFn				**O**
OEM Commands	[ICMB]	Bridge	C0h-FEh	O / No
Other Bridge Commands				**O**
Error Report	[ICMB]	Bridge	FFh	O / No

© 2007 Kontron AG

AdvancedTCA®[10]	PICMG® 3.0 Table			**M**
Get PICMG Properties	3-9	PICMG	00h	M / **Yes**
Get Address Info	3-8	PICMG	01h	N/A
Get Shelf Address Info	3-13	PICMG	02h	N/A
Set Shelf Address Info	3-14	PICMG	03h	N/A
FRU Control	3-22	PICMG	04h	M / **Yes** [2]
Get FRU LED Properties	3-24	PICMG	05h	M / **Yes**
Get LED Color Capabilities	3-25	PICMG	06h	M / **Yes**
Set FRU LED State	3-26	PICMG	07h	M / **Yes**
Get FRU LED State	3-27	PICMG	08h	M / **Yes**
Set IPMB State	3-51	PICMG	09h	N/A
Set FRU Activation Policy	3-17	PICMG	0Ah	N/A
Get FRU Activation Policy	3-18	PICMG	0Bh	N/A
Set FRU Activation	3-16	PICMG	0Ch	N/A
Get Device Locator Record ID	3-29	PICMG	0Dh	M / **Yes**
Set Port State	3-41	PICMG	0Eh	N/A
Get Port State	3-42	PICMG	0Fh	N/A
Compute Power Properties	3-60	PICMG	10h	N/A
Set Power Level	3-62	PICMG	11h	N/A
Get Power Level	3-61	PICMG	12h	N/A
Renegotiate Power	3-66	PICMG	13h	N/A
Get Fan Speed Properties	3-63	PICMG	14h	N/A
Set Fan Level	3-65	PICMG	15h	N/A
Get Fan Level	3-64	PICMG	16h	N/A
Bused Resource	3-44	PICMG	17h	N/A
Get IPMB Link Info	3-49	PICMG	18h	N/A
AMC	AMC.0 Table			
Set AMC Port State	Table 3-27	PICMG	19h	O / **Yes**
Get AMC Port State	Table 3-28	PICMG	20h	O / **Yes**

[1] Response byte 2: hours, byte 3: minutes after module start. Bytes 4..6: void

[2] Request byte 3: = only 04h (quiesced) implemented.

[3] Has oem extensions. Please refer to 6.2.2.1, Get Device ID Command with OEM Extensions

 © 2007 Kontron AG

6.2.2 OEM Commands and Extensions

6.2.2.1 Get Device ID Command with OEM Extensions

	LUN	NetFn	CMD
Get Device ID command with OEM extensions	00h	App = 06h	01h

	Byte	Data Field
Request Data	-	-
Response Data	1	Completion Code
	2:12	Regular Get Device ID Command response fields
	13	Release index (e.g. 101 denotes release index 1.01) Please note: Some tools offer this as “SDR version”.
	14	Module Geographical Address (site number): 1 … 8 = Module in AMC bay A1, A2 ,A3, A4, B1, B2, B3, B4 or in µTCA slot 1 … 8 with bus addresses 72h, 74h, 76h, 78h, 7ah, 7ch, 7eh, 80h 9 …12 = Module in µTCA slot 9 … 12 with bus addresses 82h, 84h, 86h,88h 0, > 12 = Module position is not in range. The IPMB-L bus is switched off (ref. to 8.3).
	15	Reserved
	16	Reserved

© 2007 Kontron AG

6.2.2.2 Set Firmware Parameters

The command below permits the selection of interrupts to be used during KCS communication.

	LUN	NetFn	CMD
Set Firmware Parameters	03h	OEM = 3Eh	05h

	Byte	Data Field
Request data	1	Reserved B4h
	2	Reserved 90h
	3	Reserved 91h
	4	Reserved 8Bh
	5	Cmd Flags [6:2] Reserved [1] 0b = get only [0] 0b = do not reset
	6	Operating Modes [7:5] Reserved [4] Reserved [3:1] Reserved [0] Reserved (=1b)
	7	IRQ number FFh = do not use interrupts 0Ah = use IRQ10 0Bh = use IRQ11 Any other values Reserved.
Response data	1	Completion code
	2	Cmd Flags
	3	Operating Modes
	4	IRQ number

© 2007 Kontron AG

6.2.2.3 Set Control State (Firmware Hub, Boot Order)

	LUN	NetFn	CMD
Set Control State (Firmware Hub, Boot Order)	00h	OEM = 3Eh	20h

	Byte	Data Field
Request data	1	Control ID 00h: MMC Flash selection 9Dh: Firmware Boot Order Configuration
	2	Control State for MMC Flash selection: (These settings are stored in EEPROM and applied (to logic) each time the MMC detects power-on) 00h = Flash selection by switch is not inverted 01h = Flash selection by switch is logically inverted Please note that this selection will be automatically toggled during a failing boot process. Other settings may additionally modify Control State for Boot Order Configuration: 00h .. FFh = Selected Boot Order Configuration being defined for the Net Boot Loader (refer to the Net Boot Loader documentation).. (These settings are stored in EEPROM and applied (to logic) each time the MMC detects power-on)
Response data	1	Completion code

© 2007 Kontron AG

6.2.2.4 Get Control State (Firmware Hub, Boot Order)

	LUN	NetFn	CMD
Get Control State (Firmware Hub, Boot Order)	00h	OEM = 3Eh	21h

	Byte	Data Field
Request data	1	Control ID 00h Firmware MMC Flash selection 9Dh Firmware Boot Order Configuration
Response data	1	Completion code
	2	Current Control State (refer to 6.2.2.3, Set Control State (Firmware Hub, Boot Order)

© 2007 Kontron AG

6.2.2.5 Set or Get FRU Sub Device Number

For the specific organization of the AM4100's FRU data in sub devices refer to 8.4.2, Board Specific FRU Data Organization.

For e.g. FRU data updates it is possible to temporarily override the internal FRU sub device selection by the OEM command below. Every time when officially FRU 0 data is accessed in reality the data of the selected sub device is accessed.

	LUN	NetFn	CMD
Set or Get FRU Sub Device Number	00h	OEM = 3Eh	22h

	Byte	Data Field
Request data	1	Sub device number 00h…03h: Set these numbers to select the FRU 0 sub devices 0…3. 0xFE: Set this code to get the number of the currently selected FRU 0 sub device. 0xFF: Set this code to reset the number of the currently selected FRU 0 sub device to the hardware settings (Dip switch or solder jumper, refer to the hardware manual).
Response data	1	Currently selected FRU 0 sub device number

Please note: This command selects only the storage of the currently used FRU data and **NOT** the board's operation mode! Settings changed with this command are volatile and only valid until the next reset of the MMC.

Example:
If you want to update the FRU data memory for mode 1, PCIe End Point you should proceed like this:

- Use the OEM command 'Set or Get FRU Sub Device Num' and select sub device 1.
- Use ipmitool [1] to write the FRU data via FRU 0: 'ipmitool fru write 0 <filename>'
- Use the OEM command 'Set or Get FRU Sub Device Num' and select sub device 0xff to re-activate the default sub device again.

© 2007 Kontron AG

6.2.2.6 OEM Module Quiescence Feedback

Overview:

Please refer to: 10. Hot Swap

This command provides support to control a graceful shut down of the AM4100. There is a Graceful Reboot and Shutdown Daemon (grnsd) for Linux included in the Linux BSPs being offered by Kontron. It works as described in the following (refer to “Usage for a self written shut down daemon” below).

Usage for a self written shut down daemon:

This command normally is used during a shut down to inform the MMC about the end of shutdown. If a timeout time must be **set** to avoid endless waiting for the end of payload shutdown, then the daemon sets this timeout in the command after system start with the “set quiesce wait timeout” bit set and the “Quiesce wait timeout” time <> 0. Afterwards the daemon has to call this command cyclically with the “OS daemon present” bit set. When the MMC gets a FRU Control (Quiesce) request from the carrier (e.g. during a Hot Swap sequence) it sets the “quiesce request (FRU Control)” bit in its command response. After the daemon sees this bit set in the response it should shut down the system. After having set the “quiesce request (FRU Control)” bit the MMC starts the timeout timer (if a timeout time was defined) and monitors the sleep signal line to recognize the sleep state which should be caused by the shut down. When the MMC receives a command with the “quiescence acknowledge” bit set or the timeout timer has expired, the MMC sends a “Module Hot Swap event” message to the carrier, and in the following the payload power will be switched off.

If no daemon is announced as present:

If no command call announces that a daemon is present, the MMC automatically uses the default wait timeout time 0 (endless wait) during the Hot Swap process until the daemon sends the “Os ready” using control bits = 40h (see below). But if the wait timeout time was set to a value 1…255 by control bits = 80h (see below) this wait timeout time will be used in any case while waiting for end of shutdown. The waiting for the sleep process (signal) can immediately be ended by setting “Os ready” using control bits = 40h (see below).

 © 2007 Kontron AG

	LUN	NetFn	CMD
OEM Module Quiescence Feedback	00h	OEM = 3Eh	40h

	Byte	Data Field
Request data	1	Control bits: [7] - 1b = set quiesce wait timeout [6] - 1b = quiescence acknowledge (OS ready) [5] - 1b = OS daemon present [4:0] Reserved
	2	Quiesce wait timeout [sec] a) An OS daemon is present (refer to bits above): This is the maximum time from the moment on that the MMC receives FRU Control (Quiesce) request until it sends back the appropriate Module Hot Swap event message. b) No OS daemon is present (refer to bits above): This is the maximum time from the moment on that the MMC receives FRU Control (Quiesce) request until it sends back the appropriate Module Hot Swap event message. If sleep state is recognized before timeout then the Module Hot Swap event message will be sent immediately. If the time is set to 0 (default after reset) then the Module Hot Swap event message will only be sent after recognition of sleep state (signal).
Response data	1	Completion code: 00h
	2	Control bits: [7] - Reserved [6] - 1b = quiescence acknowledge (OS ready) [5] - 1b = OS daemon present [4] - 1b = quiesce request (FRU Control) [3] - Reserved [2] - 1b = graceful reboot request (FRU Control) [1] - 1b = quiescence reached (MMC acknowledge) [0] - 1b = module hot swap switch opened
	3	Quiesce wait timeout (valid only if OS daemon present = 1)

Settings changed with this command are volatile (in particular quiesce timeout and OS daemon present). Bits [6:5] are always settable, but once the quiesce request comes they cannot be cleared until quiescence state is entered and exited. **Please note** that some carriers reset the MMC during hot swap. That means that all settings have to be renewed after that.

© 2007 Kontron AG

7. Sensors Implemented on the AM4100

The MMC includes many sensors for voltage or temperature monitoring and various others for pass/fail type signal monitoring.

Every sensor is associated with a Sensor Data Record (SDR). Sensor Data Records contain information about the sensors identification such as sensor type, sensor name, sensor unit. SDRs also contain the configuration of a specific sensor such as threshold/hystheresis, event generation capabilities that specifies sensor behavior. Some field of the sensor SDR are configurable through IPMI v1.5 command and are set to built-in initial value. Finally one field which is the sensor owner must reflect the module addresses that allow the AMC Carrier to identify the owner of the SDR when it is scanned from the module management controller and merged within the AMC Carrier Device SDR repository.

From IPMI perspective, the MMC is set up as a satellite management controller (SMC). It does support sensor devices, and uses the IPMI static sensor population feature of IPMI v1.5. All SDRs can be queried using Device SDR commands to the MMC.

The sensor name in its SDR has a name prefix which after module insertion is automatically adapted to the physical position of the module in a carrier or in a µTCA chassis. The format of this prefix is:

- in AMC bay 1…8 or µTCA slot 1…8: ‘A1:’, ‘A2:’, ‘A3:’, ‘A4:’, ‘B1:’, ‘B2:’, ‘B3:’, ‘B4:’.
- in µTCA slot 9…12: ‘R9:’, ‘Ra:’, ‘Rb:’, ‘Rc:’.

Please note that in the case that the module is installed elsewhere, then the IPMB-L address of the module is unknown and the interface is off.

Module sensors that have been implemented are listed in the sensor list below.

 © 2007 Kontron AG

7.1.1 Sensor List

For OEM (Kontron) specific sensor types and codes in the following table please refer to chapter 7.1.2.

SENSOR NAME	SENSOR TYPE (CODE) / EVENT/READING TYPE (CODE)	Ass. Mask / Deass. Mask / Reading Mask	DESCRIPTION	Causes red Health LED on error
IPMI Info-1	OEM Firmware Info 1 (C0h) / OEM (70h)	0003h / 0000h / 7FFFFh	For internal use only	N
IPMI Info-2	OEM Firmware Info 2 (C0h) / OEM (71h)	0003h / 0000h / 7FFFFh	For internal use only	N
IPMI Watchdog	Watch dog (23h) / Sensor-specific	010Fh / 0000h / 010Fh	Watchdog 2	Y
FRU Agent	OEM (C5h) / Discrete (0Ah)	0140h / 0000h / 0147h	FRU agent	N
ModuleHotSwap	OEM (F2h) / Sensor-specific	0007h / 0000h / 0007h	Hot swap sensor	N
IPMBL State	OEM (C3h) / Sensor-specific	000Fh / 0000h / 000Fh	State of IPMB-L bus	N
Storage Err	Management subsystem health (28h) / Sensor-specific	0002h / 0000h	Storage error	N
Board Reset	OEM (C4h) / Sensor-specific	04DEh / 0000h	Board reset event	Y
Board 3.3V	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	Board 3.3V supply	Y
Board 3.3vIPM	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	AMC Management Power (MP) 3.3V	Y
Board 12.0v	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	AMC Payload Power (PWR) 12V	Y
Board 1.8V	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	Board 1.8V supply	Y
Board Vtt0.9V	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	DDR termination supply	Y

© 2007 Kontron AG

SENSOR NAME	SENSOR TYPE (CODE) / EVENT/READING TYPE (CODE)	Ass. Mask / Deass. Mask / Reading Mask	DESCRIPTION	Causes red Health LED on error
Board 5.0V	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	Board 5V supply	Y
Board 1.5V	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	Board 1.5V supply	Y
Board 1.1V	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	Board 1.1V supply	Y
Temp CPU	Temperature (01h) / Threshold (01h)	0280h / 3280h / 1818h	CPU die temperature	Y
Temp CPU In	Temperature (01h) / Threshold (01h)	0280h / 3280h / 1818h	Inlet temperature near CPU	Y
Temp AMC Out	Temperature (01h) / Threshold (01h)	0280h / 3280h / 1818h	Outlet temperature near AMC connector	Y
Temp AMC In	Temperature (01h) / Threshold (01h)	0280h / 3280h / 1818h	Inlet temperature near AMC connector	Y
Temp CPU Out	Temperature (01h) / Threshold (01h)	0280h / 3280h / 1818h	Outlet temperature near CPU	Y
CPU status	Processor (07h) / Sensor-specific (6Fh)	0002h / 0002h / 0002h	CPU thermal alarm sensor	N
POST Value	OEM Post Value (C6h) / OEM (78h)	0000h / 0000h / 00FFh	POST Value (from host I/O port 80h)	N
FWH0 Boot Err	Boot Error (1Eh) / Sensor-specific	0008h / 0008h / 0008h	Firmware Hub 0 (Boot Flash 0) boot error	Y
FWH1 Boot Err	Boot Error (1Eh) / Sensor-specific	0008h / 0008h / 0008h	Firmware Hub 1 (Boot Flash 1) boot error	Y
Pwr Good	Power supply (08h) / OEM (77h)	0000h / 0000h / 18B7h	Status of all power lines	N
Pwr Good Evt	Power supply (08h) / OEM (77h)	0000h / 18B7h / 18B7h	Power fail events for all power lines	Y

© 2007 Kontron AG

SENSOR NAME	SENSOR TYPE (CODE) / EVENT/READING TYPE (CODE)	Ass. Mask / Deass. Mask / Reading Mask	DESCRIPTION	Causes red Health LED on error
Lan Front0 Lk	LAN (27h) / Sensor-specific	0000h / 0000h / 0003h	LAN Front link status or LAN port 8 status (if supported by the module)	N
Lan Front1 Lk	LAN (27h) / Sensor-specific	0000h / 0000h / 0003h	LAN port 9 status (if supported by the module)	N
Lan AMC0 Link	LAN (27h) / Sensor-specific	0000h / 0000h / 0003h	LAN port 0 status	N
Lan AMC1 Link	LAN (27h) / Sensor-specific	0000h / 0000h / 0003h	LAN port 1 status	N
Health Error	Platform Alert (24h) / Digital discrete (03h)	0000h / 0000h / 0003h	Aggregate states (power, temperatures etc.). Visualization by the Health LED.	Y
MMC Reboot	Platform Alert (24h) / Digital discrete (03h)	0002h / 0000h / 0003h	MMC reboot active state. Is asserted during boot time.	N
Ver change	Version change (2Bh) / Digital discrete (03h)	0002h / 0000h / 0003h	Version change	N

© 2007 Kontron AG

7.1.2 OEM Event / Reading Types

OEM SENSOR TYPE (CODE)	OEM EVENT/READING TYPE (CODE)	DESCRIPTION	
Firmware Info 1 (C0h)	70h	Internal Diagnostic Data	
Firmware Info 2 (C0h)	71h	Internal Diagnostic Data	
Board Reset (C4h)	6Fh (sensor type specific)	Sensor-specific Offset	Event
		00h	Reserved
		01h	HwPowerReset
		02h	PCIReset
		03h	HwWatchDogReset
		04h	SoftReset
		05h	Reserved
		06h	ColdReset
		07h	IPMICommand
		08h	Reserved
		09h	Reserved
		0Ah	BMCWatchdog
IPMBL State (C3h)	6Fh (sensor type specific)	Sensor discrete State	Meaning
		08h	IPMB-L running
		others	IPMB-L not running
Post Value (C6h)	6Fh (sensor type specific)	Sensor discrete State	Meaning
		Bits [7:0]	Post Value (read from host I/O port 80h)
		Bits [15:8]	Reserved

© 2007 Kontron AG

OEM SENSOR TYPE (CODE)	OEM EVENT/READING TYPE (CODE)	DESCRIPTION	
i.e. for Power Good / Power Good Event	77h	Sensor-specific Offset	Event
		0h	12V good
		1h	5V good
		2h	3V3 good
		3h	Reserved
		4h	1V8 good
		5h	1V5 good
		6h	Reserved
		7h	vccCore good
		8h	Reserved
		9h	1V1 good
		Ah	Reserved
		Bh	3V3IPMI good
		Ch	0V9 good
Hot swap sensor (F2h)	6Fh (sensor type specific)	Sensor-specific Offset	Event
		00h	Handle close
		01h	Handle open
		02h	Quiesced

© 2007 Kontron AG

8. Firmware Code

8.1 Structure and Functionality

MMC firmware code is organized into boot code and operational code, both of which are stored in a flash module. Upon an MMC reset, the MMC executes the boot code and performs the following:

- Self test to verify the status of its hardware and memory.
- Performs a checksum of the operational code.
- Communicates with the Firmware Upgrade Manager (FWUM) in order to inform the MCC watchdog that the actual MMC firmware is suitable for execution. Upon successful verification of the operational code checksum, the firmware will jump to the operational code.

Operational code is upgradeable in-the-field.

8.2 Firmware Upgrade

The standard way to upgrade the MMC's operational code is to use the open tool 'ipmitool' [1] together with an image file. Ipmitool allows the downloading ('ipmitool download …') and activation ('ipmitool upgrade') of the new operational code and saves an existing one. The rollback to the formerly running operational code is possible as well ('ipmitool rollback').

All IPMI interfaces which are offered by ipmitool are usable (KCS interface, IPMB bus, LAN). This allows local upgrade or remote upgrade.

Files which contain an image of operational code have the module type (AM4100) and the string "FWUM" in its name.

During the download process the currently running operational code operates as usual until the upgrade command is issued. During the upgrade process the MMC is off line for about 45 seconds when the boot code is re-organizing the firmware storage and afterwards starts the new operational code. If this doesn't succeed after a time out the boot code performs an automatic rollback to the last working operational code.

 © 2007 Kontron AG

8.3 Firmware / Module Identification

There are two ways to verify by means of IPMI that the MMC resides on an AM4100.

1. The response on the IPMI command “Get Device ID” offers among others the following response data:
   - Manufacturer ID = 3A98h (Kontron IANA ID)
   - Device ID = 04h (H8S2166)
   - Product ID = 4100
   - Firmware Revision in bytes 4:5 - depends on running firmware.
   - The SDR revision in byte 13 (OEM part of the response) is a sub revision of the firmware revision
2. The Device ID String which can be found by reading the Management Controller Device Locator Record (SDR Type 12h) contains the string “AM4100”. For a module being placed in bay B3 the Device ID String will be “B3: AM4100”.

8.4 FRU Information

8.4.1 Structure and Functionality

The MMC provides 4 kB non-volatile storage space for FRU information. Some of the data stored there, like the Module Current Requirements record or E-Keying information (refer to AMC.0 specification for details), are mandatory for module functionality in the ATCA/AMC environment. Refer to [2] Platform Management FRU Information Storage Definition document for details).

Please note that missing FRU information will prevent the AMC module from being accepted by the carrier controller during the Hot Swap process and the module will get no payload power.

Full low level access to read or write a module’s FRU Information is provided by regular IPMI FRU Device commands. Please be careful when writing FRU information directly using standard IPMI commands. Damaging the FRU Information may lead to a non working payload.

To avoid this danger there exists a Kontron Linux tool ‘frum’ (refer to section 12.2), which allows to display and partially modify FRU data. E.g. the ‘frum’ tool makes it easy to modify Product Info Area fields (like Product Version or Product Serial Number).

 © 2007 Kontron AG

8.4.2 Board Specific FRU Data Organization

The AM4100 board comes in two different hardware versions concerning its GbE interfaces

- 2 * GbE on the front panel, 2 * GbE on the AMC connector
- 4 * GbE on the AMC connector

The version is fixed during the manufacturing process.

Each of these two hardware versions can operate in four different ways:

- PCIe Root Complex (sub-device 0)
- PCIe End Point (sub-device 1)
- sRIO Host (sub-device 2)
- sRIO Agent (sub-device 3)

The code for the selection of one of these four modes of operation is adjustable by DIP switches or solder jumpers (please refer to the hardware manual). The 4 kB wide non-volatile FRU 0 data storage of the board is internally divided into four 1 kB wide sub devices 0…3. For every mode of operation the appropriate sub device is selected automatically but is offered to the outside always as FRU 0.

For FRU data updates it is possible to temporarily override the internal FRU sub device selection by the OEM command ’Set or Get FRU Sub Device Number’ in chapter 6.2.2.5. Every time when officially FRU 0 data is accessed, in reality the data of the selected sub device is accessed.

8.4.3 Download

Normally the user doesn't need to download the FRU information because the module is shipped with the data which describes best the module’s capabilities and requirements. Remember that FRU information might be damaged by a write access to the IPMI FRU Device using standard IPMI commands.

If needed the standard way to download FRU information to the module is to use the open tool ‘ipmitool’ [1] for the download of an image file (e.g. ‘ipmitool fru write 0 <file name>’).

All IPMI interfaces which are offered by ipmitool are usable (KCS interface, IPMB bus, LAN). This allows local upgrade or remote upgrade. Note that the KCS interface is only usable on a powered payload.

8.5 E-Keying

E-Keying has been defined in the AMC.0 R1.0 Specification to prevent module damage, prevent misoperation, and verify bay connection compatibility. The FRU data contains the AMC point-to-point connectivity record as described in Section 3.9.2 of the AMC.0 R1.0 specification. The Set/Get AMC Port State IPMI commands defined by the AMC.0 specification are used by the carrier for either granting or rejecting the E-keys.

 © 2007 Kontron AG

8.6 PCI Express Clock Source

The PCI-E reference clock may be generated locally by the module (default configuration) or externally (common/external reference clock connected via the AMC connector). The standard FRU E-Keying data for the AM4100 contains an AMC Link Descriptor for the PCI-E clock.

The common reference clock provided by the carrier may be slightly modulated (Spread Spectrum Clock - SSC). Standard FRU E-Keying data for AM4100 contain two AMC Link Descriptors for the PCI-E channel, first the one for SSC and second the one for none SSC.

If an external clock is available the carrier's IPMC assigns its presence during E-keying by "Set AMC Port State" commands". As a result of any SSC or non SSC selection the module will connect to the external clock by setting a signal. This connection will be activated by the next payload power on and can't be switched off until the next payload power off.

© 2007 Kontron AG

9. NetBootLoader Flash Selection

When the AM4100's payload starts, the first code to be executed is the NetBootLoader. There are two Flash devices, number 0 and 1, which may contain different NetBootLoader code. Which one of them will be selected for the **next** boot process on depends on:

a. The position of an on board DIP switch (refer to the AM4100 User Guide).

b. The decision of the MMC to logically modify (toggle) or not to modify this preselection. The MMC's current decision is stored in its EEPROM and is valid from the next boot process on.

c. The contents of a user writeable register (refer to the AM4100 User Guide).

There are two reasons for the MMC to adjust its decision or not. The first cause is an OEM IPMI command "Set Control State". The second cause is a failing boot process. Please refer to the chapters below.

9.1.1 Boot Flash Selection by OEM IPMI Command

The OEM IPMI command "Set Control State" (refer to 6.2.2.3) adjusts whether the MMC modifies the switch based Flash selection or does this not from the **next** boot process on. The MMC stores this decision in the EEPROM.

9.1.2 Automatic Boot Flash Selection during the Boot Process

After each payload reset the MMC waits for a special message from the Boot Loader. This message contains the checksum report, i.e. it reports whether the boot Flash's checksum is right or wrong. If the checksum is wrong or the message is not received within 20 seconds, then the currently used NetBootLoader Flash is assumed to contain an invalid or a corrupted image. In this case the MMC toggles again its decision to modify the preselection of the boot Flash from the next boot process on and stores this decision in its EEPROM. Then it issues a "Boot Error (Invalid boot sector) event" for the failed NetBootLoader Flash and sets the appropriate sensor value (sensor 'FWHx Boot Err'. x = 0..1). Afterwards it causes a new payload reset. When a timeout error is recognized and the count of boot errors exceeds 2 or when a checksum error is recognized and the count of boot errors exceeds 4 the MMC gives up, i.e. causes no more payload resets after a boot error until the next payload power on event.

 © 2007 Kontron AG

10. Hot Swap

As a hot-swappable field replaceable unit (FRU), the AM4100 also follows the same stringent carrier grade RASM feature set, namely - Reliability, Availability, Serviceability, Maintainability. When offered in combination with AdvancedTCA platforms, TEM (Telecom Equipment Manufacturers) clients literally conserve valuable system AdvancedTCA system slots. The AM010 supports **Full Hot Swap** capability as per PICMG 3.0. It can be removed from or installed in the system while it is on (without powering-down the system). Please refer to the PICMG 3.0 specification for additional details.

During Hot Swap of a working module the payload side must be shut down automatically ("graceful shutdown") on command of the MMC and the end of shut down must be signalled back to the MMC.

Requirements:

- At system start on payload side the Kontron shutdown daemon 'grnsd' must be started. It is included in the Linux board support packages for the AM4100 (refer to [3]). This daemon communicates cyclically with the MMC for the exchange of states, commands and acknowledges. For this it uses the "OEM Module Quiescence Feedback" command. Refer to 6.2.2.6.

Part of the Hot Swap 0peration sequence to be processed by MMC and OS:

- After the handle switch has been pulled, the carrier controller lets the MMC set a "shut down request" flag for the daemon.
- The grnsd daemon reads this request flag in the response to its cyclical "OEM Module Quiescence Feedback" command. After having read that flag, the grnsd daemon initiates the shut down of the payload software system.
- At the end of the shut down process, the grnsd daemon informs the MMC by setting the appropriate flag when calling the "OEM Module Quiescence Feedback" command.
- When the MMC reads this flag, it reports to the carrier controller that the payload shut down has finished so that the Hot Swap processing can be finished.
- By default the MMC waits endlessly for this flag. If an endless wait is to be avoided, it is necessary to set a timeout time. For the setting of the timeout refer to 6.2.2.6, OEM Module Quiescence Feedback,. After this timeout time the MMC reports to the carrier controller that the payload shut down has finished so that the Hot Swap processing can be finished.

© 2007 Kontron AG

11. Module Not Running on an ATCA Carrier

If you run the AM4100 in an environment which is not compatible to an ATCA carrier you will have to consider the following terms and conditions:

- The MMC's management power must permanently be available after system start.
- An ATCA compatible E-Keying process must be performed to cause the module to select the external PCIe clock signal. After the E-Keying, a payload power on (12V) has to be performed to activate this selection. For this the IPMI command "ChassisControl" may be used to cause a power cycle or – better – the first power up.
- There are other signals which must be set by the MMC according to the selection of run modes (mode switches). These signals only become valid after a payload power on event. So at least perform a payload power on reset after the MMC has done the start up actions. For this the IPMI command "ChassisControl" may be used to cause a power cycle or – better – the first power up.
- To get a perfect synchronization of MMC and payload it is needed that a payload power on event is performed **after** management power on. Otherwise no automatic reboot of the payload after a boot error is performed.

 © 2007 Kontron AG

12. OS Support / Tools

12.1.1 Linux

Normally all drivers and kernel modules needed for communication between the payload sided software and the MMC firmware via the KCS interface come with the distribution. Newest sources can be downloaded from http://openipmi.sourceforge.net. There may be downloaded the OpenIPMI project as well. The OpenIPMI library package includes some applications and the needed libraries. One of the applications is 'ipmicmd' which makes it possible to send and receive raw IPMI sequences; another, the 'ipmi_ui', provides a higher level interface and thus it does not require deep IPMI knowledge from the user side.

Another very useful all-in-one tool is the ipmitool (http://ipmitool.sourceforge.net). It provides a user friendly interface to many IPMI features and extensions, for example to PICMG LED control and for the upgrade of the MMC's firmware ("ipmitool fwum…").

There is a Kontron Modular Computers' IPMI ToolKit which contains some tools for the customer for the monitoring and the maintenance of some IPMI functionalities.

The following command line tools are included in the IPMI Tool Kit:

- frum: display and modification of FRU data
- temptool: selection, display, and storage of temperature and voltage sensor values

All these tools are OpenIPMI based. This toolkit is available on the "AMC Kit CD". If this CD was not shipped with your module then please contact our support (support@kontron-modular.com).

Please refer to the manual "MAN_LIN_IPMI_TOOLKIT_0103.pdf" being included in the package.

© 2007 Kontron AG

13. IPMI Communication LEDs

There are three IPMI communication LEDs on the face plate.

LED 0 (Hot Swap)
Color: Blue
Position: Bottom right
Labeled: ./.
Meaning: Indicates the Hot Swap state of the inserted and powered module.
Meaning: On: Module may be extracted.
Blinking: Hot Swap active, don't extract
Off: Module in normal operation. Don't extract.
Index in "Get/Set LED State" commands: 0

LED 1 (Out Of Service)
Color: Red
Position: Top left
Labeled: by a crossed out plus sign
Meaning: Indicates the "Out Of Service" state of the powered module.
Behavior: Solid red = module out of service
Off = module powered/running
Pulsing = Traffic on the IPMB-L bus
Index in "Get/Set LED State" commands: 1

LED 2 (Health)
Color: Green / red
Position: Top right
Labeled: by a plus sign Meaning: Indicates the "Health" state of a running/powered module.
Behavior: Blinking: Module MMC is running, showing its heart beat.
Pulsing: KCS interface active.
Off: Module MMC not running.
Any action, green: No health error detected (refer to sensor "Health Error ", 7.1.1).
Any action, red: Health error detected.
Index in "Get/Set LED State" commands: 2

 © 2007 Kontron AG